A LARGE Data Warehouse

- 30,000 users, 174+ applications
  - Any question on any data from any user anytime (within security and privacy constraints)
  - Enterprise data model – thousands of tables
  - Exceeding 300K queries/day
    - 60% < 1 second
    - 95% < 1 minute

- 296+2 nodes
  - 1016 - Intel CPUs
  - 1192 GB RAM Memory
  - 242 TB raw disk – 10,864 drives
  - 105 TB Max Perm addressable
  - ~36 GB/sec interconnect bandwidth
  - ~41 GB/sec I/O bandwidth
  - >650TB/day max physical I/O
Enterprise Data Warehouse

- **Enterprise Data Warehouse:**
  - An Enterprise Data Warehouse is a historical repository of detailed data used to support the decision-making process throughout the organization. It spans multiple subject domains and provides a consistent view of data objects used by various business processes throughout the on-line enterprise environment.

- **Data Warehousing:**
  - Data Warehousing is a process of building the data warehouse and leveraging information gleaned from analysis of the data with the intent of discovering competitive enablers that can be employed throughout the enterprise.

- **Traditional Data Warehousing** focuses on reporting and extended analysis:
  - What happened
  - Why did it happen
  - What is expected to happen next
What is Traditional Data Warehousing?

Traditional Data Warehousing is...
... an integrated & logically consistent store of detailed data used to aid the decision making process.
- Little to no interaction with customer or supplier channels
- Primarily batch data feeds
- Many “standard” reports...

What is Traditional Data Warehousing?

In addition, Traditional Data Warehousing is...
- Ad-hoc queries in support of strategic decisions...
- ... but these questions are typically asked by a business analyst based on their own experience and intuition.
- In some cases, the questions are asked too late to take profitable action...

Hmm... I wonder why I'm losing high margin customers?

13 Platinum customers were affected in Sept.
What is Active Data Warehousing?

Active Data Warehousing is traditional DW plus...
... very current detailed data (combined with historical data) for strategic, tactical, & event driven business decision making.

- Timely updates - close to real time
- Pre-designed triggers & queries designed to detect significant events
- Event Notification Services (ENS)
- Assisted decision making via analytic applications
- Tracking of tactical decision and actual results

Bob Smith is a PLATINUM CUSTOMER and will miss his connecting flight!!

What Makes a Warehouse "Active"?

- **Inter-Active**
  - Integrated customer channels
  - Integrated supplier channels
  - Integrated data analysis

- **Re-Active**
  - Manage inventory
  - Manage product cycles
  - Manage costs

- **Pro-Active**
  - Event Notification Services (ENS)
  - Automated marketing campaigns
  - Automated pricing
  - Automated replenishment

Your current rate of production will not meet the forecasted seasonal demand
Active Data Warehouse

- Active Data Warehouse (ADW):
  - A repository of detailed data required to support:
    - Strategic decision-making: Long range decisions covering broad domains
    - Tactical decision-making: Short term decisions focused on a narrow topic
    - Event based decision-making: Decisions made as a result of an event
  - Tactical decision support often requires data freshness service levels that are much more aggressive than strategic decision support.
  - This more up-to-date data is integrated with historical data in the active data warehouse.
  - Data spans multiple subject domains and provides a consistent view of data objects used by various business processes throughout the online enterprise environment.

Active Data Warehouse

- Active Data Warehousing:
  - A process of building the active data warehouse
  - Leveraging information gleaned from analysis of the data with the intent of providing assisted predictive analysis
  - Delivering actionable information to decision-making agents (human or software) on a near real-time basis.
  - Automation of business processes and decision-making, where appropriate, through the use of event detection and software based business rules.
Active data warehousing moves all analysis into the database to answer complex business questions quickly and with scalability...

Businesses need to perform analysis for planning, forecasting, profiling, fraud detection, trending, and pattern analysis to identify the proper action based on business drivers.

Strategic Decision Making

Tactical Decision Making

Active data warehousing is also about supplying information to front-line decision makers...

Businesses need repeatable, consistent execution of data-driven decisions by all constituencies, regardless of their number.
Traditional vs. Active Data Warehouse

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Traditional</th>
<th>Active</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query Types</td>
<td>Strategic &amp; Ad Hoc queries only</td>
<td>Tactical, Strategic, and Ad Hoc queries</td>
</tr>
<tr>
<td>Granularity of Time</td>
<td>Temporal granularity is coarse. Typically on the order of days.</td>
<td>Temporal granularity is fine. Typically on the order of seconds or minutes.</td>
</tr>
<tr>
<td>Usage Capacity Attributes</td>
<td>Limited number of concurrent users and / or concurrent queries.</td>
<td>Very large number of concurrent users and / or concurrent queries.</td>
</tr>
</tbody>
</table>
| Usage Attributes   | Power users, Knowledge workers, & Internal users.                            | Traditional users, plus...  
|                    |  
|                    | • Customers (via touch-points or portals)  
|                    | • Suppliers (via B2B brokers or portals)  
|                    | • Automated applications  
|                    | • Autonomous Agents                                                          |  
| Application Attributes | Periodic Report Generation       | Traditional Applications, plus...  
|                    | Ad-hoc queries used to answer “new questions.”                               |  
|                    |  
|                    | • Deep analysis of data  
|                    | • Optimization of quantitative models  
|                    | • Event driven notification & action                                         |  
|                    | • Rapid decision making                                                      |  
| ROI                | ROI is measured in course time units                                        | Measurable ROI based on easily quantified business metrics            |

An Integrated, Centralized DW Solution

[Diagram of an integrated, centralized data warehouse solution with various components and data flows]
Enterprise Data Warehouse Evolution

- Query complexity grows
- Workload mixture grows
- Data volume grows
- Schema complexity grows
- Depth of history grows
- Number of users grows
- Expectations grow

Data Sophistication

Measure

- Chasm from static to dynamic decision-making

Operationalizing

- What is happening?

Actuating

- Make it happen!

Initiate

- Continuous Update & Time-Sensitive Queries Become Important

Predicting

- What will happen?

Execute

- Event-Based Triggering

Data Warehousing – The “Must Remember” List

- A data warehouse is a solution to a business problem not a technical problem
- The warehouse needs to constantly overcome obstacles that are as yet undefined
- "Mores Law": more users wanting more applications that have more complex and varied analysis against more data with more frequent updates in a more timely manner.
- The goal behind the warehouse is consistency and agreement, not just access
- The foundation put in place determines the speed, and duration of the business evolution
Warehousing is not about Data...

- It’s about the revenue you will never bill or the opportunity you will never see
- It’s about the power to negotiate with:
  - customers, suppliers
  - competitors, regulators
- It’s about your power to maximize your assets
  - Understanding what actions drive highest returns
  - Taking actions while the value is still available
- It’s about seeing your enterprise business processes for the first time
  - Reduce costs, inefficiencies, and confusion
  - Simplify, optimize and enable

You’ve never seen your business like this before

Dimensions of the Data Warehouse Environment
Dimensions of the Data Warehouse Environment

**Corporate Maturity**
- Business Focus

**Business Governance**
- BI Requirements
- Funding Of DW Initiatives
- Prioritization of BI Initiatives
- Measurement of ROI

**Architectural Governance**
- BI Architecture
- Platform and Database Selection
- Data Architecture

**BI Workload Profile**
- Creation of BI Output
- Data Loading
- Interactions to External Systems
- Service Levels

**User Access**
- Users with Access to Data
- Reporting Medium or Tool

**Decision Support**
- Data Availability
- Analytical Capability
- Customer Profiling Ability

**Data Quality**
- Data Quality Monitoring
- Data Quality Resolution
- ECTL Tool Usage

**Data Currency**
- Data Freshness
- EAI Technology Support

**Metadata**
- Metadata Content
- Metadata Storage/Retrieval Method

**Communications and Training**
- Comm. on availability and use of BI
- BI Training
- BI Support

---

**Teradata Warehouse**

**Third-Party Tools**

- **Extract, Transform, Load**
  - Enterprise and System Tools
  - DataMart Warehouse Suites
  - Knowledge Discovery Data Mining
  - Information Access OLAP
  - Query & Reporting

- **Relational OLAP**
  - MicroStrategy
  - Business Objects

- **Client OLAP/Reporting**
  - Cognos
  - Hummingbird
  - Brio Technologies
  - Information Builders

- **MOLAP/HOLAP**
  - Hyperion
  - Essbase
  - SAS
  - Cognos

- **Tools**
  - ETI
  - Hummingbird
  - Informatica
  - Trillium
  - Ab Initio
  - DataStage xE/390
  - BEZ
  - BMC
  - Ambeo
  - Computer Associates
  - SAS
  - Informatica
  - SPSS/ISL
  - Quadstone
  - SAS
  - Information Builders
  - Hummingbird
  - SAS
Architectural Principles & Components

Toto.Horvli@Teradata-NCR.com
November 10th 2004

You've never seen your business like this before.

Why an Enterprise Architecture?

There’s a new way of defining industry leadership: “The enterprise that has the largest variety and number of important customers is the de facto leader.” This requires fast-paced changes, complexity and relationship shifting.

Gartner The Evolving Requirements for the ‘New Enterprise’, December 2002

The RTE is an enterprise that competes by using up-to-date information to progressively remove delays to the management and execution of its critical business processess

Gartner The Gartner Definition of Real-Time Enterprise, October 2002
Business Issues and Drivers

In a new competitive environment organizations need to deal with the following business issues and drivers:

- **Increasing speed to market**
  - Reduced product lifecycles
  - Reduced length of differentiation
  - Increased competition

- **Attracting and retaining loyal customers**
  - Improved buyer information
  - Increased buyer choice
  - Increased buyer expectations

As a consequence organizations need to focus on:

- **Planning and designing the strategy for the business**
  - Flexible organization
  - Business and Technology Governance
  - Change Management

- **Using Technology for competitive advantage**
  - High volume of data/information
  - Real-Time decision-making processes
  - Innovation Management

- **Focusing on core competencies**
  - Skills Management
  - People Management
  - Knowledge Management
Enterprise Architecture

**Definition - Architecture**
- The discipline dealing with the principles of design and construction
  - Webster
- The art or science of building
  - Webster

**Definition - Architect**
- A (qualified) person who designs new buildings and who makes certain that they are built correctly
  - Cambridge Dictionaries
- A person skilled in the art of building; one who understands architecture or makes it his occupation to form plans and designs of buildings, and to superintend the artificers employed
  - Webster

Enterprise Architecture Major Components

- **Enterprise Users:** Any person or application element that accesses the enterprise IT infrastructure is defined to be an "enterprise user." In the context of this paper, we define four classes of users — consumers, suppliers, and internal. For any given user class there are a number of access styles that might be used. For example, consumers might interface with the enterprise via POS terminals, self-serve terminals, web browsing, customer service representatives, call center personnel, etc. Suppliers might interface with the enterprise via specialized client server tools, web based tools, or paging devices. Internal users will typically use client server tools, web based tools, or paging devices. Trading partners typically use specialized application models based on EDI, but are slowly moving to a web based model for B2B interaction.

- **Business Specific Services:** The Business Specific Services are comprised of a set of application components. Example services include OLTP, Finance, Logistics, etc. Legacy application services are based on transaction processing systems, and client server application models. Newer applications are based on Application Brokerage Services, such as Web Services, Corba, or .NET, to name a few.

- **Application Specific Repositories:** The Application Specific Repositories are comprised of a set of specialized database repository systems that provide the system of record necessary to run the enterprise. These systems are highly tuned to maximize response time and throughput.

- **Decision-making Services:** The analytic and decision-making services perform analysis on a combination of the historical data and recent data to provide broad based intelligence that drives the business processes of the enterprise.
Enterprise Architecture Major Components

- **Decision-making Repository**: The decision-making repository stores the historical record of the enterprise. Data freshness in the decision-making repository is based on service level needs dictated by the business processes.

- **Data Acquisition Environment**: These elements provide for the acquisition of data from data generating repositories and other information systems. Data transformation, and the loading of that data into the decision support repositories is performed by a special set of applications. Data and information is moved between these environments at intervals based on the service level requirements of the firm.

- **Service Brokers**: Service Brokers are specialized software components that provide for the dispatching of service requests from clients to application service provider(s). The communication layer used by service brokers may employ low level TCP/IP protocols, or higher level internet technologies. Web Services, .NET, and CORBA are three examples of Service Brokers. Web HTTP Servers are those software components that receive HTTP requests from web browsers (and other web applications) and deliver web content in the form of HTML or XML. Web based business services are applications that provide specific business functionality (e.g. customer affinity, dynamic marketing offers, etc) and deliver the information using web protocols, such as HTML or XML.

- **Enterprise Integration**: A collection of middleware products that enable a diverse range of computing services to interact with each other. Integration can occur at the data level, message level, application level, business process level, and can include business automation. XML plays a key role in the integration of applications by providing application neutral format for the exchange of data. EAI components are found in application services, application brokerage services, data acquisition services, and user interface environments.

Enterprise Users

- **Consumers/Customers** are any persons or firm that buys goods or services from another firm. We identify a limited category of customers below.
  - **Retail**: A retail customer is defined as any "person" that buys consumer goods from a retail outlet (i.e., store). The definition of a customer is not confined by the method in which a product is purchased. A retail customer may buy goods directly at a store, through a call center, or via a web interface. In either case the person making the purchase is referred to as a "retail customer."
  - **Corporate**: A corporate customer is defined as any "business firm" that buys or services from a supplier. A corporate customer may purchase goods or services directly at a store, through a call center, or via a web interface. In either case the firm making the purchase is referred to as a "corporate customer."

- **Suppliers** are firms that sell goods or services to another firm. We identify a limited category of suppliers below.
  - **Retail**: A retail supplier is defined as any "firm" that supplies consumer goods to a retail outlet for sale to the final consumer.
  - **Corporate**: A corporate supplier is defined as any "firm" that supplies goods or services to a corporate consumer.
  - **Trading Partner**: A trading partner is a special category of "supplier" found in the B2B environment. A firm that sells a large amount of raw goods or sub-components that are used to make a larger product is a "supplier" in the trading partner relationship.

- **Internal users** are individuals or business process agents that access the enterprise IT infrastructure as part of their assigned job duties. A Sales person uses the IT services in the process of selling goods or services for a firm. A marketing person uses the IT services to analyze the market space and derive new offers. A finance person will use the IT services to determine the best way in which to deploy the financial resources of the firm. A manufacturing person will use the IT services to during build-to-order processes and manage inventory.
Computing Platforms

Various physical components and platform services needed to run repositories and applications

- **CPU**: The hardware component(s) that interprets the programs
- **Memory**: The hardware component(s) that temporarily stores data used by the CPU
- **Storage**: The hardware component(s) that provide a persistent store of data used by applications
- **Network Infrastructure**: The physical communications devices (e.g., routers, hubs, cable), embedded firmware used to control the network devices
- **Operating System Services**: This element provides the software control of the physical platforms and presents an industry standard interface to application that wish to access the resources. It provides other software functions, such as multi-tasking, shared memory, etc
- **Internet Delivery Services**: The software subsystems that sit on top of the network infrastructure and provide standard Internet services (e.g., TCP/IP, FTP, SMTP, HTTP). In practice, both an Internet and an intra-net use the same technology, and thus are viewed as synonymous in this discussion

Database Management Systems

Database Management Services

- **Data Management**
  - Base Types
  - Complex Types
  - Binary Types
  - Large Objects
  - Semi-Structured Data

- **Data Models**
  - **Data Definition**: A Data Definition Language (DDL) offers a method to describe a conceptual schema for a database. The DDL is used when a database is designed, created, and when the underlying data model is modified. The schema describes the format of data records, data types in the records, and how one record relates to one or more other records in the database
  - **Data Manipulation**: A Data Manipulation Language (DML), or query language, offers a method to insert, update, or delete records in a data set. Any modification to a database must adhere to the ACID properties:
    - **Atomicity**: All of the updates that belong to a transaction must be applied to DBMS, or none of them must be applied.
    - **Consistency**: A transaction must create a new and valid state of the DBMS. If a transaction aborts, then the DBMS must return the resources to their prior state. Referential integrity rules that are in place must not be violated.
    - **Isolation**: While a transaction is in progress, its work must be kept separate and isolated from other transactions in progress on the DBMS.
    - **Durability**: Once a transaction has been committed, its updates must be permanent and visible to all subsequent transactions on the DBMS
Session Control: The DBMS should provide services that allow multiple users or applications to establish a session with the DBMS. A session allows a user to perform multiple DML statements that are independent of each other. The DBMS needs to maintain isolation between two or more concurrent sessions.

Concurrency Control: The activity of coordinating the actions of processes that operate in parallel, access shared data, and therefore potentially interfere with each other.

Security Services: The DBMS should manage access to data according to a set of rules declared at data definition time. The rules should allow the partitioning of data into subsets that may or may not be accessed by a given user, user profile, or role. The service should allow a user to "logon" to the DBMS and perform multiple DML statements.

Data Protection: The DBMS should provide a set of services to preserve the content of the DBMS on off-line media. The database must be in a transactionally consistent state if the backup copy is to have any value. Furthermore, the DBMS should have a service that allows a DBMS to read the content from the off-line media and restore the state of the DBMS to that which was written to the off-line media.

Replication Services: In addition to backup and restore of data from the DBMS, Replication Services are used to create and manage one or more duplicate copies of data from a DBMS.

Session Control Components: The session control components should have the flexibility to communicate over a number of standard protocols, such as TCP/IP, SNA, Token-Ring, IBM-Channel Connect, etc. Session control components process user logon requests, user authentication, user authorization, etc.

Parsing Engine: The parsing engine parses commands from a user and converts the user command language (e.g., SQL) into an internal data structure that represents the various command steps necessary to fulfill the user request. There are two command types processed by the parsing engine — Data Definition Language (DDL) and Data Modification Language (DML) commands.

Dispatcher: The dispatcher accepts the execution tree as input and dispatches task requests to various subsystems in the DBMS. The dispatcher coordinates activities of multiple tasks and uses concurrency control commands to commit or rollback commands that have been issued.

File System: The File System manages the disk resources for the DBMS. It determines which disk resource will be used to store a given data object. It determines what session has control of a given data object.

Administrative Tools:
- Load Tools
- Archive & Restore Tools
Database Management Systems

DBMS Architectural Styles

- Hierarchical DBMS
- Network DBMS
- Relational DBMS
- Object Relational DBMS

An Application Specific Database is a persistent data store used by business service applications. These data stores typically contain only that data necessary to complete the set of business processes for which they are responsible. In practice, these data stores are implemented using one of the following architectural styles: Relational Database Management Systems (RDBMS), hierarchical data stores (e.g., IMS), and occasionally even flat files. These systems are usually highly tuned for transactional work loads, such as debit-credit, which have fairly predictable IO patterns. While they can service multiple concurrent users and generate reports, they are not typically well suited for performing deep analysis of data. These databases have a write oriented workload with a focus on minimizing latency on individual transaction commit times for many concurrent insert/update/delete operations.

An Enterprise Data Warehouse is a centralized store of detailed enterprise data, spanning multiple subject domains. It represents an integrated version of all the facts known to the enterprise. Detailed data is collected from a myriad of source systems and moved to the EDW once and only once. A key point to understand in the ADW architecture is that each of the source systems in the distributed environment continue to serve their unique business mission without disruption from the data capture process. The primary use of the EDW is to provide support for the quantitative and analytic decision-making process of the enterprise. Thus, the EDW must be able to store vast quantities of data. Many commercial firms need to compare business metrics on year-over-year basis. As a result, the EDW needs to persist a historical record of well over one year of detailed data. Some firms are finding value in storing and analyzing data for even longer periods — up to five years.
**Database Management Systems**

**DBMS Usage Styles**

- **An Operational Data Store** is a subset of detailed data containing recent data — usually no more than 3 to 6 months’ worth of the most recent data. The ODS provides database services in support of operational reporting and tactical decision-making.
  - A call center might use the ODS to monitor call traffic. Based on analysis of the data, it can optimize the resources of the call center based on business goals.
  - In another example, an ODS might be deployed in a call center that handles customer service for a catalogue based retailer. In this scenario, the call agents need fast access to detailed data about the customer and their recent transactions. In addition, the call agents may need to modify information about the customer (e.g., a new address) or a recent transaction (e.g., change the color of a shirt that was recently ordered, but not yet shipped).
  - A web-based application may be interacting with a customer and use an ODS to obtain state information about a customer relationship. Armed with predefined business rules and a deep understanding of the customer and current situation, it will determine if there are any marketing offers that the firm wishes to present to the customer.

- **A Data Mart** is a subset of enterprise data providing application-specific analysis for a focused group of associates — usually representing a specific department or functional area within an enterprise. The number of associates operating in the sub-group can be arbitrarily large or small, and they can be centrally located or geographically dispersed throughout the enterprise. Data is periodically extracted from various source systems and loaded into the DMs. Numerous DMs are deployed throughout the enterprise in close proximity to end users. A set of application services access the DM and aid the users in their daily activities. However, end users cannot easily perform multi-domain analysis, because they have a limited view of the enterprise data.

- **A Metadata Repository** is a specialized repository that maintains information regarding various pieces of enterprise data, such as source data, ETL processes, enterprise warehouse data, business applications, work flow, and more. The lineage (e.g., data source, data & time of creation or extraction, transformation processes, etc.) of data is maintained as well as the semantic meaning of the data. Interrelationships between applications and data repositories is usually maintained.

---

**Applications**

**Business Specific Application Services**

- Point of Sale
- Order Taking
- Service Tracking
- Fulfillment
- Billing
- Receivables
- Inquiry
- Order Placing
- ....
Applications
Decision-Making Application Services

- **Report Generation**: One of the first set of services employed in a simple decision-making environment is to generate reports that describe "what" has happened in the last reporting period. Reports about sales, revenue, costs, margin, etc. are generated. Reports can take the form of green-bar hard-copy, or may take the form of e-mail, or web pages.

- **Correlation Services**: The next set of services used in the decision-making environment produce a correlation relationship that describes what events tend to occur together (e.g. clustering, segmentation). Frequently, ad-hoc queries are used as part of these services.

- **Pattern Detection Services**: Once we know what events happen together, we can try to determine a cause-effect relationship that describes "why" something happened. This service usually takes the form of Pattern Matching, Batch Scoring, etc.

- **Automated Decision Services**
  - On-Line Scoring
  - Rules Based Services

- **Event Management Services**

Information Evolution in Data Warehousing

- **Stage 1**: Reporting, What happened?
- **Stage 2**: Analyzing, Why did it happen?
- **Stage 3**: Predicting, What will happen?
- **Stage 4**: Operationalizing, What is happening?
- **Stage 5**: Active Warehousing, What do I want to happen?

- Primarily Batch with Pre-defined Queries
- Increase in Ad Hoc Queries
- Analytical Modeling Grows
- Continuous Update & Time Sensitive Queries Gain Importance
- Event Based Triggering takes hold

- Batch
- Ad Hoc
- Analytics
- Continuous Update / Short Queries
- Event-Based Triggering
Decision Services and the 5 Stages of ADW

<table>
<thead>
<tr>
<th>Services</th>
<th>S1</th>
<th>S2</th>
<th>S3</th>
<th>S4</th>
<th>S5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Report Generation</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>OLAP &amp; Ad-Hoc</td>
<td></td>
<td>●</td>
<td>●</td>
<td></td>
<td>●</td>
</tr>
<tr>
<td>Pattern Detection &amp; Batch Scoring</td>
<td></td>
<td></td>
<td>●</td>
<td>●</td>
<td></td>
</tr>
<tr>
<td>Automated Decision Services</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>●</td>
</tr>
<tr>
<td>+ On-Line Scoring</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>+ Rules Based actions</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Event Management</td>
<td></td>
<td></td>
<td></td>
<td>●</td>
<td>●</td>
</tr>
</tbody>
</table>

Applications
Decision-Making Application Components

- Business Intelligence Query Tools
- Report Generation Tools
- Rules Engines
- Scoring Engines
- Data Mining Tools
- Middleware Services
  - Data Access Middleware
    - ODBC
    - OLE-DB
    - JDBC
Applications
Application Styles

- 2-Tiered
- 3-Tiered

- **Remote Procedure Call (RPC):** The RPC computing model provides a method for a "client" to invoke a procedure in an independent executable object, referred to as the "service provider". It is important to understand that a "client" and a "server" are software components. The client and server components may be co-located in the same computing platform, or they may execute on different computing nodes that are connected via a network.

- **Client / Server (C/S):** C/S computing describes a relationship between two computing elements that communicate with each other to complete a task. The "client" is a consumer of a service and the "server" is the producer of a service. A distinguishing feature of the C/S model is that the client always initiates an action and the server responds to the request with some form of response. The client and the server communicate with each other via an agreed upon protocol. Although the client server concept can be implemented with client and server components that operate within a single computer, the typical implementation is for the client and server components to operate on different physical computers that are inter-connected via a network.

Applications
Application Styles

- **Brokered Applications: Web Services**

  - **Service Provider:** From a business perspective, this is the owner of the service. From an architectural perspective, this is the platform that hosts access to the service. It has also been referred to as a service execution environment or a service container. Its role in the client-server message exchange patterns is that of a server.

  - **Service Requestor:** From a business perspective, this is the business that requires certain function to be satisfied. From an architectural perspective, this is the application that is looking for and invoking or initiating an interaction with a service. The requestor role can be played by a browser driven by a person or a program without a user interface, e.g. another web service. Its role in the client-server message exchange patterns is that of a client.

  - **Service Broker:** This is a searchable set of service descriptions where service providers publish their service descriptions. The service discovery agency can be centralized or distributed. A discovery agency can support both the pattern where it has descriptions sent to it and where the agency actively inspects public providers for descriptions. Service requestors may find services and obtain binding information (in the service descriptions) during development for static binding, or during execution for dynamic binding. For statically bound service requestors, the service discovery agent is in fact an optional role in the architecture, as a service provider can send the description directly to service requestors. Likewise, service requestors can obtain a service description from other sources besides a service registry, such as a local file system, FTP site, URL, or WSIL document.
Web Service Examples

- Customer Offer Service: This service produces a set of marketing offers based on customer profile. When the service "starts" it "publishes" its interface and service definitions to the broker.

- Call Center Agent: A call-center agent receives a call from a customer. Using the enterprise intra-net, the agent accesses a web page (i.e., URL) and is presented with a web-form. The agent submits the customer-id to the web form, which is forwarded to a Java application.

- The Java application "finds" the "customer offer" service via the broker. The Java application "interacts" with the "customer offer" service. Messages take the form of XML messages over the intra-net.

- The Java application transforms the newly obtained information so that it can be forwarded to the agent's web browser. Data is sent to the agent's browser as an XML file with an associated XSL file. Information is visualized on the agent's screen using the presentation services offered by the browser.

Java Services
Data Integration: Integration at the data level is core to the EAI mission. The goal of data integration is to make data from one application accessible to another application. If data from a secondary application is to be useful, the syntax and semantics of the data must be well defined. Understanding both the physical database structure, as well as the business rules used by the "owning" application are imperative to the goal of data sharing.

Message Integration: Message level integration refers to the integration of applications at the message level. This level of integration is a fundamental enabler for the new architectural style of web based applications. These applications are "loosely coupled". That is to say there is a loose binding at the application interface level. Services and their interfaces are discovered at run time. The application designer does not know specific message layout at design time. XML is the technology that makes this possible.

Application Integration: Integration at the application level expands on data level integration by facilitating access to business logic contained in enterprise applications. More sophisticated multi-tier architectures use "application middleware" to provide an infrastructure for message passing, message queuing, object brokering, and/or web service brokering. In addition various "adapters" may be used to bridge applications designed to operate in two different environments.

Business Process Integration: Business Process Integration (BPI), refers to the integration of two or more business processes to complete a higher level business activity. The integration of sequential business processes is referred to as "Work Flow". Work Flow integration takes the output of a business process and uses that information as input to the next business process. Any number of business processes can be linked together to achieve a single objective.

Data Access Middleware provides a standard method for accessing relational and non-relational data stores, regardless of their underlying architecture, and the proprietary syntax used for access. Well know products that fall into this category include; ODBC, JDBC, etc.

Remote Procedure Call (RPC) provides a mechanism for a software component to invoke a procedure that exists in a second component (i.e. external to the first component). This is how Client-Server computing began. This is a synchronous protocol. The client is blocked until the server responds.

Message Oriented Middleware provides an infrastructure for passing messages between two or more software components. This is the next evolution of Client-Server computing. This protocol can be either synchronous or asynchronous.

Message Queuing extends the capabilities of other middleware by allowing a message to be stored in a queue for deferred processing (i.e. asynchronous processing). Data is stored on "reliable" media (i.e. reliability). Transational semantics (e.g., "get", "put", "commit"). Guaranteed delivery.

Object Request Brokers (ORB) are distributed infrastructures that allow binary objects to interoperate, regardless of the underlying platform environment, or the development environment (i.e. C, C++, JAVA, etc.) used to create the object.

Transaction Managers (TM) are used to coordinate transactions across distributed heterogeneous data sources. The Open Group has developed a standard model, referred to as the "XA Protocol" that allows components from different vendors to interoperate in a distributed transaction.

Web Services are refer to a collection of applications and middleware that use internet technologies to send messages and access services. Messages are transported over internet protocols, such as HTTP. Java and XML allow browsers and web enabled applications to interact with each other using standard interfaces and protocols.
The "publish & subscribe" model is a connectionless message passing system. It is used in environments where events occur fast and frequently. For example, a stock exchange might use a P&S system to publish price quotes to end users that are monitoring the market on a real time basis.

Peer-to-peer (P2P) is a communications model in which each element has a set of similar capabilities and either party can initiate a communication session. In some cases, peer-to-peer communications is implemented by giving each communication node both server and client capabilities. In recent usage, peer-to-peer has come to describe applications that employ the Internet to exchange files with each other directly or through a mediating server.

Electronic Data Interchange (a.k.a. Electronic Document Interchange) (EDI) is defined as the electronic exchange of standardized business documents over a communications network linking two or more trading partners. Documents types include purchase orders, invoices and shipping manifests and are intended for application-to-application processing. A key element of EDI is the Value Added Network (VAN) — a privately owned network that manages the connections between participating companies, supplies additional services, like trading partner agreements (TPA), and oversees the network.

Publish and Subscribe
**Active Data Warehousing**

**Events**

- **Event Occurrence**: An event occurs at a single point in time.

- **Event Detection**: An event is eventually detected by the enterprise. Detection of the event occurs at a single point in time. Most enterprises will log the event and forward a notification of the event to one or more application services for analysis.

- **Analysis**: Once an event occurs, one or more application service components will analyze a set of variables associated with the event. The number of variables can vary from a handful to hundreds of variables. Frequently the ADW will have created a “score” prior to the event that is used to determine the best course of action. In other cases, application components in the Decision-Making Services will dynamically score the event. Dynamic scoring is one example of a tactical query. At the culmination of this step, the impact of the event is assessed and a recommended course of action is derived.

- **Action**: After the recommended course of action is determined, one or more decision-making service components will forward the prescribed action to elements in the field that need to react to the event. These elements can be either people or software agents. Information is pushed to a user or system rather than being requested by one. Initially, an enterprise will require that people perform the prescribed actions. But as the enterprise evolves and the processes of Active Data Warehousing matures, a person provides diminishing value to the business process.

- **Dissintermediation**: Dissintermediation occurs when the enterprise has enough confidence in the automated decision-making process balanced against the perceived risk of choosing an incorrect course of action. Many firms have already dissintermediated some of their processes. For example, most retailers have automated the process of replenishment of commodity products in their enterprise.

- **Logging**: The enterprise needs to record what actions were taken so that an audit can be performed posthumously to the original event. This is a critical part of any continuous business improvement process. In some situations, the actions and interactions will not exactly match those recommended by the decision-making services. Therefore, it is necessary to record the outcomes of the decisions that were made.

- **Continuous Monitoring**: Continuous monitoring is the process of comparing predicted results to observed results with the intent of refining the models used by the enterprise. During the analysis phase an analytic application derived a suggested action based upon the output of a quantitative model. These models have a degree of standard error. Nonetheless, the enterprise may be able refine the model(s) based on thorough analysis after downstream outcomes become evident.

---

**Teradata Event Management**

The figure illustrates the Teradata Event Management system, which includes components such as the Active Warehouse, Industry Data Model, Business and/or Predictive Models, Triggers, CRM, Stored Procedures, Event Engine, UDFs, and Predictive Agent. The system is designed to handle events by collecting, analyzing, and reacting to them in real-time, ensuring continuous improvement and operational efficiency.
Active Data Warehousing

Data Acquisition Rates

One of the primary goals of an Active Data Warehousing is to optimize the response made by an enterprise to a set of events. To achieve this goal requires the acquisition of very current data from many areas throughout the enterprise. Data should be acquired at a rate that maximizes its business value. In contrast to "real-time" acquisition of data, which implies virtually instantaneous data acquisition, we offer the concept of "right-time" data acquisition.

Right-time data acquisition is the process of acquiring data as fast as is necessary and no faster than is required to optimize the business value of that data. It is based on the premise that, for a given class of data, there is an optimal rate of data acquisition for the enterprise. Right-time data acquisition is a balance between the time value of data and the cost of data acquisition.
Active Data Warehousing
Information Feedback

Tactical Query: A tactical query is a database operation that attempts to determine the best course of action right now. Whereas the strategic query provides the information necessary to make long term business decision, a tactical query provides information to rank and file elements in the field that need to respond quickly to a set of unfolding events.

Strategic Query: A strategic query is a database operation that attempts to determine what has happened, why it happened, and/or what will happen next. It typically accesses vast amounts of detailed data from the warehouse and ranges in complexity from simple table scans to multi-way joins and sub-queries. Applications that generate strategic queries include; report generation, OLAP, decision support, ad-hoc, data mining, etc. While the query may access large amounts of detailed data, it frequently returns a small amount of aggregated or summarized information.

Update Query: An update query is a database operation that modifies the state of a database. TERADATA provides a set of bulk load utilities used to load large quantities of data into the a database in an efficient fashion. Update operations available in SQL include insert, update, and delete. Traditional data warehouses typically append (i.e., insert) large blocks of data to the end of “detail tables”, such as retail sales line items, or call detail records. These data load operations tend to occur in coarse grained time intervals, such as days, weeks, or months. Further, the occurrence of updates and deletes to the database are very small.
Strategic vs Tactical Queries

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Strategic Queries</th>
<th>Tactical Queries</th>
</tr>
</thead>
</table>
| High Level Questions | What has happened?  
What did something happen?  
What will happen next? | What should I do right now? |
| Scope              | Very broad — accessing vast amounts of detailed data from multiple subject domains. | Narrow in scope. Uses a small set of input variables, such as a single customer ID, or a small set of product IDs. |
| Data Access Profile | Full table scans, complex joins, sub-queries, aggregations, groupings, etc. | Small number of rows accessed.  
May use derived tables that contain pre-scored values.  
Simple logic operations. |
| Response Time Requirements | Broad range of response-time requirements; from minutes to hours to days. | Predictable response time is of paramount importance. |
| Frequency / Arrival Rate | Relatively infrequent. Frequency increases at the end of a business period, such as end-of-year, end-of-quarter, end-of-month, etc. | Higher frequency of occurrence. Because tactical queries tend to be driven by external events, such as the delay of commercial airplane, or the behavior of consumers, the frequency has a more random distribution. |
| Result set size    | The size of the result set can vary from a small number of rows to a large report with a considerable number of rows, sub-groups, aggregations, etc. | Tactical queries tend to produce a very small result set. It is not uncommon for the result set to be less than a dozen rows. Usually the result set is designed to fit into a single window on a display screen. |

Enterprise Repository Styles

*Federated Database Approach*

---

**Enterprise Users**
- Consumers
- Suppliers
- Internal
- Partners

**Internet / Intranet**

**Business Specific Services**
- Service Brokers
- EAI Message Bus

**Decision Making Services**
- EAI
- Other

**Application Specific Repositories**
- OLTP
- DLTP

---

28
Teradata is Different

- Parallelism built-in from the ground up
- Dedicated to automatic management & operation
- Easy and fast data movement in and out
- Committed to the highest levels of reliability and availability
- High concurrency mixed workload management
- Unequaled scalability in every dimension

... but still looks like any other RDBMS

- Relational Database Management System
  - Tables & columns
- ANSI SQL (and more)
- Third Normal Form (3NF)
A Sample Teradata System

Snaps into your Operation
Database Architecture

- **Teradata System**
  - Users
  - Databases
  - Hierarchies

- **Teradata Objects**
  - Tables
  - Views
  - Indices
  - Triggers
  - Macros
  - Stored Procedures
  - Functions

Index types

- **Primary Index**
  - Unique
  - Non-Unique
  - Used for row distribution by hash

- **Secondary Index**
  - Unique
  - Non-Unique
  - Ordered (Range)
  - Covered (Hash)
  - Used for faster data access

- **Hash Index**
  - An alternative to single-table join

- **Join Index**
  - Index structure for derived table set caused by join, also known as materialized view

- **Partitioned Primary Index**
Analytical Functions/OLAP

- SQL-99 Window Functions
- Multiple Aggregate Distincts
- Sampling
- Recursive SQL
- Teradata specific functions for doing complex analytical querying and data mining are:
  - RANK - (Rankings), QUANTILE - (Quantiles), CSUM - (Cumulation)
  - MAVG - (Moving Averages), MSUM - (Moving Sums)
  - MDIFF - (Moving Differences), MLINREG - (Moving Linear Regression)
  - ROLLUP - (One Dimension of Group), CUBE - (All Dimensions of Group),
  - GROUPING SETS - (Restrict Group), GROUPING - (Distinguish NULL rows)

Traditional Task Centric & Data Centric

**Traditional Task Centric**
(SMP Model)

- Uniform and shared access to all platform resources (disk, etc) is **REQUIRED**

**Data Centric**
(Distributed Model)

- Exclusive access to a subset of resources
Traditional Task Centric Design

- Software design assumes that any time an application has direct access to any resource:
  - Storage
  - Memory-based resources (tables, etc)
- IMPACT: Platform hardware and OS MUST provide uniform access to ALL resources from every CPU
- Concurrent processes compete for the common pool of resources:
  - Applications coordinate access among themselves
  - Lock Manager
  - Only a subset of resources are involved in each query at any given time
- Overwhelming majority of all software today is built using this model:
  - Traditional model of software process architecture taught in schools today

Data Centric Software Architecture

- Software model: Uniform unit of parallelism
- Balanced collection of three abstracted platform resources:
  - Processor
  - Memory
  - Disk (Storage)
- Each physical node may support one or more Virtual Amps
- Data is equally distributed across Amps:
  - Hash Partitioned on primary index
- Each Amp owns a subset of resources:
  - Few rows of every table
  - No shared resources among the applications
- Parallel effort on every request:
  - Each AMP works only on a subset of data
  - Every AMP is involved in every query
  - Combine the results
  - Performance tracks data set size
- Overwhelming majority of software today does NOT use a Data Centric Architecture:
  - Not widely taught in schools today
  - Requires the development of a parallel optimizer
  - Typically of little benefit in very small SMP systems
  - Scaling is targeted at distributed systems
Scalable Hardware

- **Node**
  - Commodity Server components
  - Independent OS Kernel
  - OS scalability not an issue

- **Storage**
  - Independent I/O
  - Scales per node

- **BYNET Interconnect**
  - Fully scalable bandwidth
  - 120MB/sec/node

- **Connectivity**
  - Fully scalable
  - Channel - ESCON, B/T
  - LAN, WAN

- **Server Management**
  - One console to view the system

BYNET Switching Network

- Scales linearly as nodes are added
- Fault tolerant - within one net, plus dual nets
- Guaranteed delivery
- Database support functions - merge, broadcast, multicast, semaphores
Teradata RDBMS V2

Communication Interfaces

LAN Gateway  AMP1  AMP5
Channel  AMP2  AMP6

Parsing Engine Vprocs

PE1  AMP1  AMP5
PE2  AMP2  AMP6
AMP3  AMP7
AMP4  AMP8

PDE (Parallel Database Extensions)
NCR UNIX, W2K or NT Operating System

SMP Node

Inside One Unit of Parallelism
A Microcosm of the Database

VPROC 1
VPROC 2
VPROC 3
VPROC n

Reading
Writing
Building Indexes
Row Locking
Sorting
Aggregating
Transaction Journalling

Each VPROC owns and manages all database activity against its data
File System

- File system built of raw UNIX slices / NT Partitions
- Rows stored in blocks
  - Variable length
  - Grow and shrink on demand
  - Rows located dynamically; can be moved to reclaim space
  - Maximum block size is configurable
    - System default or per table
    - 6K to 64K
    - Change dynamically
- Indexes are just rows in tables

---

File System

- Blocks stored in virtual cylinders
  - No fixed addresses
  - Blocks can be moved to reclaim space
  - If block fills to max block size, it splits and becomes two blocks
  - If cylinder fills, it splits and half the blocks move to a new cylinder
  - No physical adjacency requirement, new cylinder goes wherever there is room

- Cylinders located via Master Index (MI)
  - Contains cylinder address, tableid and starting rowhash of cylinder
  - Always locked in memory

- Blocks located via Cylinder Index (CI)
  - Contains block location, tableid, starting rowhash and size of block

- Guarantees worst case 2 I/Os for a row
  - No page chains, empty pages, ...
  - CIs and blocks cached based on availability of memory and knowledge of query
File System

- Space allocation is entirely dynamic
  - No tablespaces or journal spaces or any pre-allocation
  - Spool (temp) and tables share space pool, no fixed reserved allocations
- If no cylinder free, combine partial cylinders
  - Dynamic and automatic
  - Background compaction based on tunable threshold
- Quotas control disk space utilization
  - Increase quota (trivial online command) to allow user to use more space
- No reorgs
  - Don’t even have a reorg utility
- No index rebuilds
- No re-partitioning
- No detailed space management
- Easy database & table definition
- Minimum ongoing maintenance
  - All performed automatically

Query Processing

- Complex SQL query features - an analytic engine
  - Derived tables, Case expressions, all forms of sub queries, Sample, ...
  - StdDev, Variance, Skew, Kurtosis
  - Rank, Quantile, Moving Average, Running Total
  - Big limits: 64 table joins, 64 nesting levels, ...
- High performance algorithms
  - Join, Aggregation, Sort, ...
  - Compiled expressions
**Fully Scalable Algorithms**

- **Aggregation without sort**
  - Read locally
    - Feed local aggregate cache which accumulates by group key
    - When EOF, purge local aggregate cache to global aggregator
  - Global aggregator is fully scalable/distributed
    - Hash group key, send by hash to global
    - Received into global aggregate cache, accumulate per group key
  - Fill/spill
    - When local cache fills, spill lowest hit items to global
    - If global fills, spill to disk - final merge over spill blocks

- **Ordered Analytics**
  - Rank
    - Estimate value partitions
    - Redistribute to value partitions - all amps
    - Sort locally
    - Cascade counts
    - Assign rank
  - Fully scalable
    - Others bring to one UoP to sort and rank

**Query Processing**

- **High performance views**
  - No penalty for using views
  - Enables wide use for ease of use, security, privacy

- **Wide choice of indexes**
  - Unique and non-unique secondary
  - Dynamic bit-mapping
  - Covered and value ordered
  - Join and Aggregate Join indexes
  - Partitioned Primary Index

- **Everything is parallel**
  - No single threaded operations (scalability killers)
    - Scans, Joins, Index access, Aggregation, Sort, Insert, Update, Delete
    - All schema operations: Create Table, Create Index, Collect Statistics, ...
Optimizer Intelligence Unconditional Parallelism

- Teradata’s Cost based optimizer provides unequalled ad hoc and complex query performance:
  - Fully Parallel
  - Cost-based
  - Look-ahead
  - Dynamic application to all queries
  - Auto-applies time-saving structures (Temp tables, Join Indexes)
  - No “HINTS” required
  - Parallel aware
- Parallelism is automatic
- Parallelism is unconditional
- Each query step fully ‘parallelized’
  - Scans, Joins, Index access, Aggregation, Sort, Insert, Update, Delete

Dimensions of Parallelism

- Multiple Steps are for a query executed in Parallel
  - Multi-table queries fast
- Large table rows passed to multiple queries in Parallel
  - ... Less I/O’s for large # of Users
- Multiple processes execute on different Nodes in Parallel
  - ... More I/O per Second
- Each Step is executed in Parallel
  - ... Full table Scans are easy
- Multiple users execute in Parallel
  - ... Serves larger population
- Multiple processes execute on different CPUs in Parallel
  - ... More computation per second
High Availability - Software

- Vproc migration
  - Move the work to functioning resources
  - Node Failure Protection
  - OS failure Protection
- Parallel recovery and rollback
- Online utilities
  - Load, Export, Backup
  - Purge
  - Checkpoint - Restart
- Eliminate many operations
  - Reorg, Index rebuild
- Fallback
  - Covers even catastrophic failures

Fault Tolerant Hardware

- Data center processors
  - Dual AC power, redundant power supplies
  - Redundant fans
  - Hot plug components
- Storage
  - Raid protected
  - Dual I/O components for all paths to disk
  - "Cliques" cross connection of nodes to disk arrays for failover
- BYNET Interconnect
  - Dual networks
  - Fault tolerant within each network
  - Automatic network reconfiguration
- Connectivity
  - Multiple redundant connections
Administration WorkStation AWS

- Provides centralized monitoring and control for Teradata MPP system
  - Power, Cooling, Nodes, WES Storage, BYNET
  - Operating System Log collection
    - UNIX MP-RAS: Console & Streams logs
    - W2K: Application, Security & System logs

- Remote Monitoring & Control
  - Events can be forwarded via EMAIL, Pager, SNMP & Customer Care Link
  - SNMP can be used to escalate events to Enterprise Management products such as CA Unicenter, Tivoli, BMC Patrol, etc.
  - Remote access via Command Line Interface or Window Terminal Services
    - Password protected Login
    - Can do everything remotely except physical tasks (FRU replacement, Press Node Dump Switch, etc.)

Business Continuity

<table>
<thead>
<tr>
<th>Traditional Configuration</th>
<th>High Performance System</th>
<th>Performance Continuity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Large Cliques</td>
<td>8 Nodes, 2 Clique</td>
<td>67%</td>
</tr>
<tr>
<td>Hot Standby Node</td>
<td>Enhanced Performance System</td>
<td>86%</td>
</tr>
<tr>
<td>Hot Standby Node + Large Clique</td>
<td>Extreme Performance System</td>
<td>100%</td>
</tr>
</tbody>
</table>

| Extreme Performance System | 3 Nodes, 1 HSN |
| Performance Continuity | 100% Lower TCO for Larger Systems |
ETL Tool Architectures

Code Generation Tools

In the early 1990s, most organizations developed custom code to extract and transform data from operational systems and load it into data warehouses. In the mid-1990s, vendors recognized an opportunity and began shipping ETL tools designed to reduce or eliminate the labor-intensive process of writing custom ETL programs.

Early vendor ETL tools provided a graphical design environment that generated third-generation language (3GL) programs, such as COBOL. Although these early code-generation tools simplified ETL development work, they did little to automate the runtime environment or lessen code maintenance and change control work. Often, administrators had to manually distribute and manage compiled code, schedule and run jobs, or copy and transport files.
ETL Tool Architectures

Engine-Based Tools

To automate more of the ETL process, vendors began delivering “engine-based” products in the mid to late 1990s that employed proprietary scripting languages running within an ETL or DBMS server. These ETL engines use language interpreters to process ETL workflows at runtime.

The ETL workflows defined by developers in the graphical environment are stored in a metadata repository, which the engine reads at runtime to determine how to process incoming data.

Although this interpretive approach more tightly unifies design and execution environments, it doesn’t necessarily eliminate all custom coding and maintenance. To handle complex or unique requirements, developers often resort to coding custom routines and exits that the tool accesses at runtime.

These user-developed routines and exits increase complexity and maintenance, and therefore should be used sparingly.

ETL Tool Architectures

Engine Based Vs. Code Generated Based

- **Engine Based**
  - Perform the transformations in object code typically on designated ETL UNIX and WNT/W2000 servers.
  - More popular than code generators because engine products are easier to use, and for the most part, eliminate need to post, maintain and execute code.
  - If a firm buys an engine, then it must pay license and maintenance fees for every server they implement (production, unit test, system test, etc.).
  - Metadata defined with the tool and actively used by processing components.
  - Processes cannot be touched by customer; they always perform inside the ETL product’s engine. Metadata remains valid.

- **Code Generated Based**
  - Metadata defined with the tool and used to generate procedural code in C, Java, COBOL, etc. which must be moved and executed on either the source, target or designated ETL server.
  - Gives IT control to promote code through its own defined, customized development process — unit test, system test, etc., production.
  - Code generation vendors generally do a better job of getting at mainframe data, including non-relational data, in various forms: IMS, QSAM, VSAM, DB2/MVS.
  - Source code can be modified by customer thus breaking with metadata and causing manual steps to be taken for any future modification needed.

Metadata in this situation includes the schemas of sources and targets and the transformation logic for data flows.
Approaches: ETL vs. ELT

ETL - all transformation occur prior to being loaded to database.
- Most ETL tools are designed to do transformation prior to loading to DBMS.
- Simpler processing design particularly for non-Teradata trained people who tend to think in record at a time mode.

ELT - data loaded to database where some transformations occur using SQL and the database.
- Power of Teradata utilized to perform potentially massive transformations by using SQL, Insert Selects, Stored Procedures, and Triggers
- Way of moving data into tables in stages to allow result consistency for query operations or to avoid locking issues.
- May be less expensive for our customers to add capacity to Teradata system for this workload than to have resources needed for a large SMP ETL server. This approach is certainly advantageous to Teradata.
- Hummingbird and ETI are the ETL vendors that support these types of processes. Others would require additional Teradata/RDBMS processes outside the scope of their product thus having a break in metadata.
What is Business Intelligence? 1/2

Business Intelligence (BI) is:
The processes, technologies and tools needed to turn data into information and information into knowledge and knowledge into plans that drive profitable business action.
BI encompasses data warehousing, business analytics and knowledge management
(The Data Warehouse Institute, Q4/2002)

Business Intelligence (BI) is:
defined as "knowledge gained about a business through the use of various hardware/software technologies which enable organizations to turn data into information"
(DM Review)

Turning data into information into action and results

What is Business Intelligence? 2/2

Business Intelligence (BI) is:
defined as "knowledge gained about a business through the use of various hardware/software technologies which enable organizations to turn data into information"
(DM Review)

Turning data into information into action and results
Business Intelligence OLAP Report:

FASMI,
"Fast Analysis of Shared Multi-dimensional Information"

Business Intelligence is a OLAP Research Company

FASMI – 1/2

- Fast
  Fast responses. Most responses are returned within 5 seconds. Simple responses take no more than 1 second. Few responses take more than 20 seconds

- Analysis
  Analytical capabilities. The system can cope with any business logic and statistical analysis relevant to the application and user, and keep it easy enough for the user to comprehend

- Shared
  Shared data. The system implements security requirements for confidentiality of shared data, and if write-back of the data is necessary, the system provides locking at the appropriate level
Multidimensional
A multidimensional view of the data. This includes full support of hierarchies and multiple hierarchies. Typical hierarchies include Time (year / month / week / day), Products (all products / category / subcategory / class / item), and Location (all locations / region / district / store / POS).

Information
Information availability. All the data and information needed must be available, wherever it is and however much is relevant to the application.

To stay up to date with OLAP visit website at www.olapreport.com

How Multidimensional Systems View Data

Most OLAP tools are built around a multidimensional database (MDDB)

Multidimensional systems view data from a business user’s perspective

Typically, the user wants to look at data in more than two dimensions

The multidimensional structure, or cube, is built around elements called dimensions (eg. time, location, product, organizational structure etc.)

Companies will build cubes that look at multiple dimensions at a time
Examples of Business Intelligence Activity

Ad Hoc Query & Reporting

Enterprise Reporting

Multi-Dimensional Analysis

Data Mining

What Rhymes with OLAP?

- DOLAP (Desktop OLAP)
- MOLAP (Multidimensional OLAP)
- ROLAP (Relational OLAP)
- HOLAP (Hybrid OLAP)

It’s common for companies to use more than one OLAP tool within their enterprise, and these may represent multiple OLAP architectures.
DOLAP: Desktop OLAP

- DOLAP tools generate small cubes either in advance or on the fly, which are distributed to desktops and processed locally

  - Characteristics of DOLAP:
    - Client (desktop) based
    - Prebuilt cubes
    - Smaller data sets
    - Fast performance
    - Lower cost per seat versus other models

MOLAP: Multidimensional OLAP

- MOLAP uses server-based multidimensional engines that can manage their own multidimensional databases as well as accessing relational databases. MOLAP tools are sometimes called MDDBMs (Multi-dimensional Database Management Systems)

  - Characteristics of MOLAP:
    - Server based
    - Aggregated data
    - Prebuilt cubes
    - Larger data sets
    - More users
    - Supports Web-based processing
    - Sourced from a single point (data warehouse) or multiple points (OLTP systems or data marts)
ROLAP: Relational OLAP

- ROLAP tools use server-based engines which provide multidimensional analyses working directly on top of the relational database engine

  - Characteristics of ROLAP:
    - Server based
    - Works directly on top of a relational database
    - Supports larger volumes of data
    - Cubes are built directly off the volume of data supported by the relational engine.
    - Additional flexibility through ad hoc queries into the detailed data

HOLAP: Hybrid OLAP

- In the HOLAP environment, the vendor provides a product that can support the OLAP model of choice for the customer. HOLAP allows customers to implement their solution under a MOLAP, ROLAP, or mixed mode. Customers need to be aware of performance issues with the various models

  - Characteristics of HOLAP:
    - Server based
    - Allows MOLAP, ROLAP, or mixed mode environment
    - Some performance issues
Data Mining

Toto.Horvili@Teradata-NCR.com
October 6th 2004

You've never seen your business like this before.

Data mining is a process of discovering and interpreting patterns in data to solve a business problem.

What is Data Mining?

[Diagram showing stages of data mining: Stage 1 - What happened?, Stage 2 - Why did it happen?, Stage 3 - What will happen?]
Comprehensive Data Mining Tool

Data Pre-processing
- Transformation Functions
- Data Visualization Functions
- Exploration Functions
- Data Reduction Matrices

Understand Business Goal
Understand Source Data

Analytic Model Design
- Advanced Statistical Functions
- Machine Learning Algorithms
- Project Management
- Data Visualization
- Scalable Technology

Analytic Model Deployment
- Model Integration
- Life Cycle Management

Analytic Model Deployment

Data Mining & Data Warehousing

- Data warehouse is the foundation for effective data mining
- Up to 70% of a data mining project is accessing and preparing data
- Data mining drives the need for huge enterprise volumes of detailed data
- Teradata Warehouse Miner™ was developed to move data mining functions into the Teradata database

Think about the implications ...

Enterprise Data Mining
Data Mining Applications

Typical Applications
Customer Segmentation
Propensity to Buy
Profitability Modeling & Profiling
Customer Attrition
Channel Optimization
Fraud Detection

How can I tell which transactions are likely to be fraudulent?

Detect and prevent fraud to minimize loss.

In-Database Mining

Traditional Method
- Moves the data to the tool.
  - Extracts samples to the data mining server
  - Slow, inefficient, prone to human error & compromise model accuracy

Vs.
“In-place” Data Mining
- Move Mining functions to the Data
  - Faster, efficient & mines ALL your detailed data!
  - Data Mining that scales with Teradata database!

Teradata Warehouse Miner
SQL
Data Mining & OLAP
SQL extensions

Data Management
Issues
Data Redundancy
Source
Samples Only
Data Movement
Data Analysis & Preparation Functions

- **Descriptive Statistics**
  - Univariate Statistics
    - Count/Minimum/Maximum/Mean
    - Standard Deviation
    - Standard Error
    - Variance
    - Coefficient of Variance
    - Skewness
    - Kurtosis
    - Uncorrelated Sum of Squares
    - Corrected Sum of Squares
  - Quantiles
    - Top 10/Bottom 10 Percentiles
    - Deciles
    - Quartiles
    - Tertiles
- **Value Analysis**
  - Data Types
  - Count
  - # NULL Values
  - # Positive Values
  - # Negative Values
  - # Zeros
  - # Blanks
  - # Unique Values
- **Modes**
- **Histograms**
- **Frequency Analysis**
- **Key Overlap**

Transformation Functions
- Derive Data Bin Number
- Derive Categorical Codes
  - Dummy-code numeric values
- Derive "Uncategorize" Codes
  - Re-code categorical values
- Range Scaling
- Z-Score transformation
- Sigmoid transformation
- Trigonometric Functions
  - SIN, COS, TAN
- Hyperbolic Functions
  - SINH, COSH, TANH
- Exponent
- Logarithms (base 10 and Natural)
- Power
- Ranks
- Quantiles
- Moving Averages
- Moving Sums
- Cumulative Sums
- Moving Differences
- Moving Least Square Linear Regression

Complex Transformations
- Product Ownership Bitmap
- Time-series Product Ownership Bitmap
- Transaction Intensity Usage Statistics
- Transaction Variability Usage Statistics

Matrix Functions
- Pearson Product-Moment Correlation Matrix
- Covariance Matrix
- SSCP Matrix
- Corrected SSCP

Data Reorganization Functions
- Random Sample
  - # Rows
  - % Rows
  - Multiple Samples
- Partition
- Denormalize
- Join
- Merge columns into results table

Analytical Techniques and Functions

- **Multivariate Statistics**
  - Linear Regression
    - Regression Model Coefficients and Statistics
    - Incremental R^2
    - Step-Wise Linear Regression
  - Factor Analysis
    - Principal Component Analysis
    - Principal Axis Factors
    - Maximum Likelihood Factors
    - Orthogonal & Oblique Rotations
  - Logistic Regression
    - Logit Model Coefficients, Odds Ratios and Statistics
    - Model Success Analysis Table
    - Step-Wise Logistic Regression
  - Visualization
    - Scatter Plot
    - Lift Chart
    - Regression Plots
    - Factor Pattern
    - Score Plot
  - Multivariate Diagnostics
    - Extensive Collinearity Diagnostics
    - Automated Identification of Constants
    - Raw level diagnostics and much more...

- **Machine Learning Techniques**
  - Decision Tree/Rule Induction
    - Gini / Regression (from CART)
    - Entropy (from C4.5)
  - Clustering
    - K-Means
    - Nearest Neighbor Linkage
    - Expectation Maximization
    - Gaussian/Poisson Mixture Model
  - Visualization
    - Graphical Tree Browser
    - Cluster Sizes / Distance
    - Cluster Measures / Membership
  - Neural Networks
    - Back Propagation
    - Experimental Prototype
    - Radial Basis Function
    - Kohonen
  - Affinity and Sequence Analyses
  - Feature Rich Implementations

- **Model Scoring**
  - SQL-Based Model Scoring
    - Decision Trees
    - Clustering
    - Linear Regression
    - Logistic Regression
    - Factor Analysis

- **Model Evaluation**
  - SQL-Based Model Evaluation
Thank you!

Proven solutions. Experience, not experiments.

www.teradata.com